UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/844,993	04/27/2001	Jacob Dreyband	033144-017	1373

23562 7590 08/02/2005

BAKER & MCKENZIE
PATENT DEPARTMENT
2001 ROSS AVENUE
SUITE 2300
DALLAS, TX 75201

EXAMINER
CHANNAVAJJALA, SRIRAMA T

ART UNIT	PAPER NUMBER
2166	

DATE MAILED: 08/02/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/844,993	**Applicant(s)** DREYBAND ET AL.	
	Examiner Srirama Channavajjala	**Art Unit** 2166	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 22 June 2005.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-36 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-36 is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)	4) □ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)	5) □ Notice of Informal Patent Application (PTO-152)
3) □ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.	6) □ Other: _____.

DETAILED ACTION

Response to Amendment

1. Claims 1-36 are presented for examination

2. Examiner acknowledges applicant's "amendment under 37 CFR 1.111" filed on 6/22/2005.

3. Claims 1,9,15,23,29 have been amended [6/22/2005]

4. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 14 Oct 2004 has been entered and a non-final Office action mailed on 12/23/2004.

5. Claims 1,9,15,23,29 have been amended [see 10/14/2004]

6. Examiner acknowledges applicants' amendment filed on 11/7/2003, paper no. 9.

7. Claims 1-6,9,11,15-20,23,25,27,29-35 have been amended, paper no. # 9.

Drawings

8. The formal drawings filed on 11/7/2003, paper no. # 10 are acceptable for examination.

Information Disclosure Statement

9. The information disclosure statement filed on 4/27/2001, paper no. # 2 has been considered and a copy was enclosed with this office action, paper no. # 5.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

10. Claims 1-3, 9, 15-17, 23, 29-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Grady et al., UML for XML schema mapping specification, 12/08/99 in view of Davidson et al., [hereafter Davidson], US Pub.No. 2002/0133811 filed on 14 Dec. 2000.

11. As to Claims 1,15,29, Grady et al., teaches a system which including 'mapping a descriptive language including a data description having a structure complexity into an object oriented programming language [see Abstract, page 2, 1.1], Grady is directed to

standard object oriented language schemas, more specifically UML for XML schema mapping as detailed in Abstract, further Grady also suggests for example object management group where UML has been established certain standards, as best understood by the examiner, descriptive language is to enhance future extensibility and reusability of information in any embedded system for example XML is one of the suitable tool as detailed in Abstract, further it is noted that Grady specifically suggested unified modeling language or UML is a standard object-oriented language that corresponds to object oriented programming language [see Abstract, line 1-3];

'receiving the data description' [page 2, item 2], Grady specifically directed to mapping data types in XML schema to classes, further Grady teaches data types semantics that are related to XML schema concept, see table in page 3;

'identifying a complex-type element in the data description' [page 3, item 1.4,page 6, item 1.8], identifying complex-type element is integral part in the XML document instances of Grady because firstly Grady is directed to XML schema,[see page 6, item 1.8], secondly, Grady specifically teaches for example defining two different data type(s) as detailed in page 3, item 1.4, further it is noted that complex types in XML schemas are user defined data types that can include other elements or attributes, complex types can contain elements defined as either simple or complex, complex types can also include attributes and groups, whereas simple types can only contain facets [see page 6, item:1.8], As best understood by the examiner, complex types are defined using the complex type element and typically contain combination of element, attribute, and group declaration, as well as references to globally declared elements and groups, further a complex type can be thought of as a mini-schema that defines the valid structure and data contained within a specific element as detailed in page 6, item 1.8;

'creating an executable object oriented class corresponding to the identified complex-type element, wherein the class includes an internal static class wherein the internal static class corresponds to the structure complexity of the data description' [page 4, 1'5, page 6 example the XML schema], as best understood by the examiner, static class analysis is a kind of data flow analysis that computes a set of classes for each variable and expression in a method is integral part of object oriented language such as C++, further it is noted that descriptive language is not only enhance future extensibility but also reusability of classes and methods, for example a simple **static class** is created as shown below and this is common knowledge object oriented environment.

```
[code]
 #include <iostream>
 class test {
 static int counter;
 public:
 int getcount() { return counter;}
 test();
 };
 int test::counter = 0;

 test::test() {
 counter++;
 }

 int main(void) {
```

```
test xyz, bar;

cout << xyz.getcount() << "\n";
}
[/code].
```

Although, creating an executable object oriented class is integral part of Grady’s teaching because firstly, Grady is directed to UML or Unified Modeling Language is itself a standard object-oriented design language [see ABSTRACT], secondly, executable UML is the next layer of abstraction depends on profile of UML, also describes the data and the behavior, further executable UML doesn’t make coding decisions, and make use of compiler to generate code, thirdly executable UML is a subset of UML proper as detailed above, on the other hand, to be useful, this subset of UML must have a way to specify actions at a higher level of abstraction than a programming language. There are many executable profiles that could be defined to select a set of meaningful components and how they interact during execution. For example, a object can invoke methods of other objects, which in turn invoke more methods is part of Grady’s Unified modeling language

It is however, noted that Grady does not specifically teach “object oriented class that is independently executable in any of a plurality of run-time environments’, although Grady specifically suggests unified modeling language or UML is a standard object-oriented design language. On the other hand, Duftler disclosed “object oriented class that is independently executable in any of a plurality of run-time environments’ [page 1, col 1-2, 0009-0014], Duftler teaches object oriented language for example Java specifically defining and implementing XML as detailed in 0011-1113. As best understood by the examiner, Duftler also specifically suggests implementing JavaBeans components using any scripting language or languages that corresponds to any of a plurality of object oriented languages executable in run-time environments.

It would have been obvious to one of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Duftler et al. into UML for XML schema mapping specification of Grady because both Grady and Duftler are directed to XML schema, and both are directed to object oriented language [see Grady: Abstract; Duftler: abstract, page 1, col 1, 0005-0006] and are from same field of endeavor.

One of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Duftler et al. into UML for XML schema mapping of Grady because that would have allowed users of Grady to define and implement object oriented language for example Java independently executable in a run-time because Java is executable in any platform, further bringing the advantages of JavaBean components automatically generated at run time [see page 1, 0006].

12. Claims 8,14,22,28,36 most of the limitations of this claim have been noted in the rejection of Claim 1 above. In addition, with respect to the claimed feature Duftler disclosed 'naming space with said internal static class to provide an implementation of said structure complexity' [see page 9, 0122-0123].

13. As to Claims 2,16,30 Grady teaches a system which including 'receiving the data description comprises receiving an XML Schema [see page 6, 1.8 XML schema]. As best understood by the examiner, the purpose of XML schema is to define the building blocks of an DML document, just like a data type definition, further it should be noted fundamental XML schema defines such as: elements that appear in a document, defines attributes that appear, defines which elements are child elements, defines the order of child elements, defines the number of child elements, defines whether an element is empty or can include text, defines data types for elements and attributes, defines default and fixed values for elements and attributes [see Grady: page 4, item 1.5]

14. As to Claims 3,17,31, the limitations of this claim have been noted in the rejection of above claim. In addition, Grady disclosed 'validating the data description ' [see Abstract, page 4 1.5 defining element type]. As best understood by the examiner, XML Schema provides powerful dedicated validation features for things like uniqueness, referential integrity, enumerations, complex types and the various data type facet as suggested by Grady, at page 3, item 1.3.

15. As to Claims 9 and 23, Grady teaches a system which including 'mapping a schema including a structural complexity into an executable object oriented programming language wherein the object oriented programming language provides a one to one correspondence between the structural complexity of the Schema and the functionality of the object oriented programming language' [see Abstract], Grady specifically directed to unified modeling language which is a standard object oriented design language that is used by the object management group, further XML schema is integrated for example developing an object model that represented in DML, describing relationships between XML and system to process it as detailed in page , introduction, Schema corresponds to Grady's XML schema as detailed in page 2, item 1.1.

Although Grady teaches, an executable object oriented language which is integral part of Grady's teaching because firstly, Grady is directed to UML or Unified Modeling Language is itself a standard object-oriented design language [see ABSTRACT], secondly, executable UML is the next layer of abstraction depends on profile of UML, also describes the data and the behavior, further executable UML doesn't make coding decisions, and make use of compiler to generate code, thirdly executable UML is a subset of UML proper as detailed above, on the other hand, to be useful, this subset of UML must have a way to specify actions at a higher level of abstraction than a programming language. There are many executable profiles that could be defined to select a set of meaningful components and how they interact during execution. For example, a object can invoke methods of other objects, which in turn invoke more methods is part of Grady's Unified modeling language;

'receiving said schema' [page 3, item 3, 1.3, section 4, page 6], schema corresponds to XML schema as detailed in section 4, page 6,;

'validating said schema' [see Abstract, page 4 1.5 defining element type],

'creating a set of executable object oriented classes including a set of internal static classes to provide a mapping of the schema into the object oriented language' [page 4, 1'5, page 6 example the XML schema], as best understood by the examiner, static class analysis is a kind of data flow analysis that computes a set of classes for each variable and expression in a method is integral part of object oriented language such as C++, further it is noted that descriptive language is not only enhance future extensibility but also reusability of classes and methods, for example a simple **static class** is created as shown below and this is common knowledge object oriented environment.

```
[code]
#include <iostream>
class test {
static int counter;
public:
int getcount() { return counter;}
test();
};
int test::counter = 0;

test::test() {
counter++;
}
```

```
int main(void) {
test xyz, bar;

cout << xyz.getcount() << "\n";
}
[/code].
```

It is however, noted that Grady does not specifically teach "independently executable in any of a plurality of run-time environments', although Grady specifically suggests unified modeling language or UML is a standard object-oriented design language. On the other hand, Duftler disclosed "object oriented class that is independently executable in any of a plurality of run-time environments' [page 1, col 1-2, 0009-0014], Duftler teaches object oriented language for example Java specifically defining and implementing XML as detailed in 0011-1113. As best understood by the examiner, Duftler also specifically suggests implementing JavaBeans components using any scripting language or languages that correspond to any of a plurality of object oriented languages executable in run-time environments.

It would have been obvious to one of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Duftler et al. into UML for XML schema mapping specification of Grady because both Grady and Duftler are directed to XML schema, and both are directed to object oriented language [see Grady: Abstract; Duftler: abstract, page 1, col 1, 0005-0006] and are from same field of endeavor.

One of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Duftler et al. into UML for XML schema mapping of Grady because that would have allowed users of Grady to define and implement object oriented language for example Java independently executable in a run-time because Java is executable in any platform, further bringing the advantages of JavaBean components automatically generated at run time [see page 1, 0006].

16. Claims 4-14,18-28,32-36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Grady et al., UML for XML schema mapping specification, 12/08/99, Duftler et al., [hereafter Dulftler], US 2002/0133811 as applied to claims 1,15,29 above, and further in view of Davidson et al., [hereafter Davidson], US Patent No. 6083276.

17. As to Claims 4,10,18,24,32, both Grady, Duftler teaches a system which including XML data description, mapping specification [Grady: see Abstract; Duftler: Abstract], however, Grady and Dulfler do not specifically teach 'mutator method', although both Grady, and Duftler suggests for example standard object oriented design language that is widely used in software development area [see Grady: Abstract; Duftler: Abstract]. On the other hand, Davidson disclosed 'mutator method' [col 24, line 65-67, col 25, line 1-7], examiner interpreting mutator method corresponds to Davidson's mutator methods as detailed in col 25, line 4-6, fig 5.

It would have been obvious to one of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Davidson et al., into UML for XML schema mapping specification of Grady et al., and Bean scripting components which is XML based language for defining and implementing JavaBean of Duftler et al. because they all are directed to XML mapping the schema [see Grady et al., Abstract, page 2: 1.1; Duftler: Abstract,;Davidson: fig 1, element 122], they all are directed to descriptive language including a data description [see Grady et al. XML example page6; Duftler: page 1, col 2, 0013; Davidson: col 8, line 10-20, col 21, line 50-65, fig 3A-4A] and they all are directed to XML environment and are both from the same field of endeavor. One of ordinary skill in the art at the time of the invention would have been motivated to combine the references with Davidson et al. because that would have allowed users of Grady's UML for XML schema mapping, Bean scripting components which is XML based language for defining and implementing JavaBean of Duftler to control which relative combinations of specific properties, events, methods, classes satisfies his or her needs as suggested by Davidson et al [col 4, line 48-58].

18. As to Claims 5,11,19,25, and 33, both Grady and Davidson teach 'validity determination as to said data description' [see Grady: Abstract, page 2, 1.1; Davidson: fig 3A-4A], Davidson teaches 'sending request including said data description from a user to a remote server' [fig 1, col 7, line 30-40].

19. As to Claims 6,12,20,26,34, the limitations of this claim have been noted in the rejection of claim above. In addition, Davidson disclosed 'reading said data description into a set of valid descriptor classes' [col 9, line 41-52], 'creating a set of objects out of the data description wherein the occurrence of an object reflects validity' [col 10, line 4-20].

20. As to Claims 7,13,21,27,35, the limitations of this claim have been noted in the rejection of claim above. In addition, Davidson disclosed 'Java, C++,Smalltalk' [col 2, line 21-29].

Response to Arguments

21. Applicant's arguments filed on 6/22/2005 with respective to Claims 1-36 have been fully considered but they are not persuasive, for examiner's response, see discussion below:

a) At page 12, claims 1,9,15,23,29, applicant argues that "specifically, Duftler merely teaches a different and arguably more efficient means for creating JavaBeans to be executed in Java code such that the JavaBeans do not have to be individually encoded, but such code is still not independently executable in a run-time environment".

As to the above argument [a], as best understood by the examiner, firstly, Duftler is directed to defining and implementing JavaBeans using XML language [page 1, col 1, 0003], secondly, Duftler also suggested that JavaBean code automatically generated at run-time [page 1, col 1, 0006, line 10-11], thirdly, Duftler suggests Bean Scripting Component or BSC that combines the concepts that including JavaBean programming and code fragments from BSC because Bean Scripting component is a XML based language [see page 1, col 2, 0011-0013]. It is further noted Duftler specifically suggested that defining and implementing JavaBeans components using any scripting language or languages

b) At page 12, claims 1,9,15,23,29, applicant argues that applicants have amended claims to recite that the claimed language creates an executable object oriented class that is independently executable in any run-time of a plurality of environments, not just in a Java environment. Thus, the present claims provide for a universally executable object oriented language, as described in the present application, and is not limited to a single language implementation.

As to the above argument [b], as best understood by the examiner, Grady specifically teaches UML, using XML schema implemented into standardized by Object management group (OMG) to behave like a programming language specific data structure [see page 1-2]. On the other hand, Duftler also teaches Beans Scripting Components or language is XML data structure page 2, col 1, 0029 defining and implementing JavaBean components using any scripting language [page 1, col 2, 0013, line 1-3], therefore, combination of Grady, Duftler suggests implementing JavaBeans components using any scripting language or languages that correspond to any of a plurality of object oriented languages executable in run-time environments.

c) At page 12-13, claims 1,9,15,23,29, applicant argues that "there is no motivation for one skilled in the art who is employing a Schema language, such as in Grady and in the present application, to combine the teachings in Duftler since Duftler merely teaches employing XML in a specific descriptive function.

As to the above argument, firstly both Grady, Duftler suggests XML schema, more specifically unified modeling language implementing XML schema [see Grady: Abstract], while Duftler specifically teaches Bean Scripting Component [BSC] is an object oriented component typically is a XML-based language defining, implementing JavaBean script. It is noted that both Grady, Duftler teach XML schema

[Grady: page 2, 1.1. XML schema and UML; Duftler: page 2, col 1, 0029], and both Grady, Duftler specifically directed to object oriented language and they both are from same field of endeavor. One of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Duftler et al. into UML for XML schema mapping of Grady because that would have allowed users of Grady to define and implement object oriented language for example Java independently executable in a run-time because Java is executable in any platform, further bringing the advantages of JavaBean components automatically generated at run time [see page 1, 0006].

Also, examiner applies above arguments to the dependent claims.

d) At page 13-14, applicant argues that neither the combination of Grady and Duftler nor the combination of Grady, Duftler and Davidson teaches or suggests all of the elements of the independent claims 1,9,15,23,29.

As to the argument [d] above, it is noted that both Grady, Duftler specifically directed to XML schema and mapping in a object oriented environment, but do not specifically teach "mutation method". On the other hand Davidson suggests "mutator method as detailed in col 24, line 65-67, col 25, line 1-7. As best understood by the examiner, mutator method is a method that changes the value of the argument variables in as detailed in col 25, line 4-6, fig 5. Therefore, it would have been obvious to one of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Davidson et al., into UML for XML schema mapping specification of Grady et al., and Bean scripting components which is XML based language for defining and implementing JavaBean of Duftler et al. because they all are directed to XML mapping the schema [see Grady et al., Abstract, page 2: 1.1; Duftler: Abstract,; Davidson: fig 1, element 122], they all are directed to descriptive language including a data description [see Grady et al. XML example page6; Duftler: page 1, col 2, 0013; Davidson: col 8, line 10-20, col 21, line 50-65, fig 3A-4A] and they all are directed to XML environment and are both from the same field of endeavor. One of ordinary skill in the art at the time of the invention would have been motivated to combine the references with Davidson et al. because that would have allowed users of Grady's UML for XML schema mapping, Bean scripting components which is XML based language for defining and implementing JavaBean of Duftler to control which relative combinations of specific properties, events, methods, classes satisfies his or her needs as suggested by Davidson et al [col 4, line 48-58].

Conclusion

The prior art made of record

a. Grady et al., UML for XML schema mapping specification published on 12/8/1999, page 1-8.

b. US Patent No. 6083276

c. US Patent No. 20020133811

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure

c. US Patent No. 5794030

d. US Patent No. 6540142

e. US Patent No. 6569207

f. US Patent No. 6418446

g. US Patent No. 6026408

h. US Patent No. 5797137

i. US Patent No. 6490581

j. US Patent No. 5956730

k. US Patent No. 5809505

l. US Patent No. 6446256

m. Lucian et al., Mapping XML and relational schemas with clio, 2 pages

n. Migrating from XML DTD to XML schema using UML, Rational Software white paper, year 2000, pages 1-8

o. US Patent No. 5794030

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Srirama Channavajjala whose telephone number is 571-272-4108. The examiner can normally be reached on Monday-Friday from 8:00 AM to 5:30 PM Eastern Time.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Alam, Hosain, T, can be reached on (571) 272-3978. The fax phone numbers for the organization where the application or proceeding is assigned is 571-273-8300 Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free)

sc
Patent Examiner.
July 18, 2005.

SRIRAMA CHANNAVAJJALA
PRIMARY EXAMINER